

Approaches for enumerating permutations with a prescribed number of occurrences of patterns

Brian Nakamura*

Abstract

In recent work, Zeilberger and the author used a functional equations approach for enumerating permutations with r occurrences of the pattern $12\dots k$. In particular, the approach yielded a polynomial-time enumeration algorithm for any fixed $r \geq 0$. We extend that approach to patterns of the form $12\dots (k-2)(k)(k-1)$ by deriving analogous functional equations and using them to develop similar algorithms that enumerate permutations with r occurrences of the pattern. We also generalize those techniques to handle patterns of the form $23\dots k1$ and derive analogous functional equations and enumeration algorithms. Finally, we show how the functional equations and algorithms can be modified to track inversions as well as handle multiple patterns simultaneously. This paper is accompanied by Maple packages that implement the algorithms described.

1 Introduction

Let $\sigma = \sigma_1 \dots \sigma_k$ be a sequence of k distinct positive integers. We define the *reduction* $\text{red}(\sigma)$ to be the length k permutation $\tau = \tau_1 \dots \tau_k$ that is order-isomorphic to σ (i.e., $\sigma_i < \sigma_j$ if and only if $\tau_i < \tau_j$ for every i and j). Given a (permutation) pattern $\tau \in \mathcal{S}_k$, we say that a permutation $\pi = \pi_1 \dots \pi_n$ *contains* the pattern τ if there exists $1 \leq i_1 < i_2 < \dots < i_k \leq n$ such that $\text{red}(\pi_{i_1} \pi_{i_2} \dots \pi_{i_k}) = \tau$, in which case we call $\pi_{i_1} \pi_{i_2} \dots \pi_{i_k}$ an *occurrence* of τ . We will define $N_\tau(\pi)$ to be the number of occurrences of τ in π . For example, if the pattern $\tau = 123$, the permutation 53412 avoids the pattern τ (so $N_{123}(53412) = 0$), whereas the permutation 52134 contains two occurrences of τ (so $N_{123}(52134) = 2$).

For a pattern τ and non-negative integer $r \geq 0$, we define the set

$$\mathcal{S}_n(\tau, r) := \{\pi \in \mathcal{S}_n : \pi \text{ has exactly } r \text{ occurrences of the pattern } \tau\}$$

and also define $s_n(\tau, r) := |\mathcal{S}_n(\tau, r)|$. The corresponding generating function is defined as

$$F_\tau^r(x) := \sum_{n=0}^{\infty} s_n(\tau, r) x^n.$$

Observe that the classical pattern avoidance problem corresponds to the case where $r = 0$ and has been well studied. In this setting, $\mathcal{S}_n(\tau, 0)$ is known to be enumerated by the Catalan numbers if $\tau \in \mathcal{S}_3$ [11, 19]. However, much is still unknown even for length 4 patterns. For example, permutations avoiding the pattern 1324 have been notoriously difficult to enumerate. Precise asymptotics are not even known, although Bóna recently gave an improved upper bound for the growth rate (in [3]) by modifying the approach used by Claesson, Jelínek, and Steingrímsson [8]. The survey

*Mathematics Department, Rutgers University-New Brunswick, Piscataway, NJ, USA. [bnaka@math.rutgers.edu]

paper by Kitaev and Mansour [10] provides an extensive overview of work in this area as well as related problems in permutation patterns.

While the more general problem (where $r \geq 0$) has also been studied, the work has usually been restricted to small patterns (usually length three) and small r . In [14], Noonan studied permutations containing exactly one occurrence of 123 and proved that $s_n(123, 1) = \frac{3}{n} \binom{2n}{n-3}$. Burstein recently gave a short combinatorial proof for the result [6]. In [15], Noonan and Zeilberger presented an approach using functional equations to enumerate $s_n(\tau, r)$ for small r and for the patterns 123, 312, and 1234. Subsequent work has been done by Bóna [5, 4], Fulmek [9], Mansour and Vainshtain [12], Callan [7], and many others. Many of these focused on finding $F_\tau^r(x)$ for $\tau \in \mathcal{S}_3$ and for small r .

One difficulty arising from the initial Noonan-Zeilberger functional equation approach in [15] was that the approach became very complicated for even $r = 2$. In addition, there are many patterns that this approach does not readily extend to. One such pattern (explicitly mentioned in [15]) is 1432. A modified approach was recently presented in [13] for the case of increasing patterns. Given a fixed $r \geq 0$, the resulting enumeration algorithm for computing $s_n(12\dots k, r)$ is polynomial-time (in n). This, in a sense, tackles the first difficulty from [15] and allows us to enumerate the sequence $s_n(12\dots k, r)$ for even larger fixed r .

In this paper, we extend the enumeration techniques in [13] to new families of patterns (including the pattern 1432) as well as multiple patterns. It should be noted that this general approach is different from the *enumeration schemes* approach pioneered by Zeilberger [21] and extended by Vatter [20], Pudwell [16, 2], and Baxter [1, 2]. The enumeration schemes approach is useful for enumerating pattern-avoiding permutations (the $r = 0$ case) but does not appear to be readily adaptable to the generalized setting for permutations with $r > 0$ occurrences of a pattern.

The paper is organized in the following manner. Section 2 extends the approach in [13] to the patterns 132, 1243, 12354, and so on. Section 3 generalizes the techniques used in [13] and applies them to the patterns 231, 2341 (which is equivalent to 1432), and so on. Section 4 extends this approach to handle multiple patterns simultaneously as well as refining by the inversion number¹. Section 5 lists some possible future work as well as some conjectures. The enumeration algorithms developed in this paper are implemented in the Maple packages `FINCR`, `FINCRT`, `F231`, `F2341`, `F123n132`, `F1234n1243`, and `FS3`. They are all available from the author's website.

2 Counting occurrences of the pattern $12\dots(k-2)(k)(k-1)$

In this section, we adapt the approach in [13] (for increasing patterns) to the patterns $12\dots(k-2)(k)(k-1)$. We first handle the case of 132 in full detail and then outline how to generalize this approach to patterns 1243, 12354, and so on.

2.1 Permutations containing 132

Given a (fixed) pattern τ and non-negative integer n , we define the polynomial

$$f_n(t) := \sum_{\pi \in \mathcal{S}_n} t^{N_\tau(\pi)}. \quad (1)$$

¹This is technically the same as tracking the number of 21 patterns that occur.

Observe that the coefficient of t^r in $f_n(t)$ is exactly equal to $s_n(\tau, r)$. For a fixed pattern τ and fixed $r \geq 0$, our goal is to quickly compute $s_n(\tau, r)$. In the remainder of this section, we will assume that $\tau = 132$.

In addition to the variable t , we introduce the catalytic variables x_1, \dots, x_n and define the weight of a length n permutation $\pi = \pi_1 \dots \pi_n$ to be

$$\text{weight}_{132}(\pi) := t^{N_{132}(\pi)} \prod_{i=1}^n x_i^{\#\{(a,b) : \pi_a > \pi_b = i, 1 \leq a < b \leq n\}}.$$

In general, this will be written more simply as $\text{weight}(\pi)$ when the fixed pattern is clear from context (in this case 132). For example, $\text{weight}(12345) = 1$, $\text{weight}(13245) = tx_2$, and $\text{weight}(25143) = t^4 x_1^2 x_3^2 x_4$.

For each n , we define the polynomial

$$P_n(t; x_1, \dots, x_n) := \sum_{\pi \in S_n} \text{weight}(\pi).$$

Observe that P_n is essentially a generalized multi-variate polynomial for f_n and in particular, $P_n(t; 1, \dots, 1) = f_n(t)$.

Given a permutation $\pi = \pi_1 \dots \pi_n$, suppose that $\pi_1 = i$. Note that $N_{132}(\pi)$ is equal to the number of occurrences of 132 in $\pi_2 \dots \pi_n$ plus the number of occurrences of 21 in $\pi_2 \dots \pi_n$, where the term corresponding to the “1” is larger than i . Letting π' be $\text{red}(\pi_2 \dots \pi_n)$, we observe that

$$\text{weight}(\pi) = x_1 x_2 \dots x_{i-1} \cdot \text{weight}(\pi')|_{x_i \rightarrow tx_{i+1}, x_{i+1} \rightarrow tx_{i+2}, \dots, x_{n-1} \rightarrow tx_n}. \quad (2)$$

This leads to the functional equation:

$$P_n(t; x_1, \dots, x_n) = \sum_{i=1}^n x_1 x_2 \dots x_{i-1} \cdot P_{n-1}(t; x_1, \dots, x_{i-1}, tx_{i+1}, \dots, tx_n). \quad (\text{FE132})$$

Once $P_n(t; x_1, \dots, x_n)$ is computed, the catalytic variables x_1, \dots, x_n can all be set to 1 to get $f_n(t) = P_n(t; 1, \dots, 1)$.

However, it is not necessary to compute $P_n(t; x_1, \dots, x_n)$ in its entirety prior to setting the catalytic variables to 1. Observe that by (FE132), we have:

$$P_n(t; 1, \dots, 1) = \sum_{i=1}^n P_{n-1}(t; 1 [i-1 \text{ times}], t [n-i \text{ times}]).$$

We get terms of the form $P_{a_0+a_1}(t; 1 [a_0 \text{ times}], t [a_1 \text{ times}])$ in the summation, which can again be plugged into (FE132) to get:

$$\begin{aligned} P_{a_0+a_1}(t; 1 [a_0 \text{ times}], t [a_1 \text{ times}]) &= \sum_{i=1}^{a_0} P_{a_0+a_1-1}(1 [i-1 \text{ times}], t [a_0 - i \text{ times}], t^2 [a_1 \text{ times}]) \\ &\quad + \sum_{i=1}^{a_1} t^{i-1} P_{a_0+a_1-1}(1 [a_0 \text{ times}], t [i-1 \text{ times}], t^2 [a_1 - i \text{ times}]) \end{aligned}$$

Now, we must deal with terms of the form $P_{a_0+a_1+a_2}(t; 1 [a_0 \text{ times}], t [a_1 \text{ times}], t^2 [a_2 \text{ times}])$. We can continue this recursive process of plugging new terms into (FE132) to eventually compute

$f_n(t) = P_n(t; 1 [n \text{ times}])$. This is much faster than the direct weighted counting of all $n!$ permutations, although it is still unfortunately an exponential-time (and memory) algorithm.

This algorithm has been implemented in the procedure `F132full(n,t)` (in the Maple package `FINCRT`). For example, the Maple call `F132full(8,t);` computes $f_8(t)$ and outputs:

$$\begin{aligned} t^{31} + 7t^{30} + 20t^{28} + 37t^{27} + 41t^{26} + 109t^{25} + 162t^{24} + 169t^{23} + 322t^{22} + 397t^{21} + 647t^{20} + 730t^{19} \\ + 1048t^{18} + 1152t^{17} + 1417t^{16} + 1576t^{15} + 1770t^{14} + 1853t^{13} + 2321t^{12} + 2088t^{11} + 2620t^{10} \\ + 2401t^9 + 2682t^8 + 2489t^7 + 2858t^6 + 2225t^5 + 2593t^4 + 1918t^3 + 1950t^2 + 1287t + 1430 \end{aligned}$$

Suppose that for a small fixed $r \geq 0$, we wanted the first 20 terms of the sequence $s_n(132, r)$. By this functional equation approach, one would compute $f_n(t)$ and extract the coefficient of t^r for each n up to 20. This approach would expend quite a bit of computational effort in generating unnecessary information (namely, all the t^k terms where $k > r$). This issue can mostly be circumvented, however, by a couple of observations. The first is the following lemma from [13]:

Lemma 1. *Let $n = a_0 + a_1 + \dots + a_s$ (where $a_i \geq 0$ for each i) and suppose $s > r + 1$. Then, the coefficients of t^0, t^1, \dots, t^r in*

$$\begin{aligned} P_n(t; 1[a_0 \text{ times}], \dots, t^{s-1}[a_{s-1} \text{ times}], t^s[a_s \text{ times}]) \\ - P_n(t; 1[a_0 \text{ times}], \dots, t^r[a_r \text{ times}], t^{r+1}[a_{r+1} + a_{r+2} + \dots + a_s \text{ times}]) \end{aligned}$$

all vanish.

Proof. The more general function $P_n(t; x_1, \dots, x_n)$ is a multi-variate polynomial. \square

This lemma allows us to collapse all the higher powers of t into the t^{r+1} coefficient and allows us to consider objects of the form $P_n(t; 1[a_0 \text{ times}], \dots, t^r[a_r \text{ times}], t^{r+1}[a_{r+1} \text{ times}])$ regardless of how large n is.

Let $n := a_0 + a_1 + \dots + a_{r+1}$. Also, for any expression R and positive integer k , let $R\$k$ denote $R[k \text{ times}]$. For example, $t^3\$4$ is shorthand for t^3, t^3, t^3, t^3 . Now for any polynomial $p(t)$ in the variable t , let $p^{(r)}(t)$ denote the polynomial of degree (at most) r obtained by discarding all powers of t larger than r . Also, define the operator CHOP_r by $\text{CHOP}_r[p(t)] := p^{(r)}(t)$.

An application of (FE132) and CHOP_r to $P_n^{(r)}(t; 1[a_0 \text{ times}], \dots, t^r[a_r \text{ times}], t^{r+1}[a_{r+1} \text{ times}])$ becomes:

$$\begin{aligned} & P_n^{(r)}(t; 1\$a_0, \dots, t^r\$a_r, t^{r+1}\$a_{r+1}) \\ &= \text{CHOP}_r \left[\sum_{i=1}^{a_0} P_{n-1}^{(r)}(t; 1\$(i-1), t\$(a_0-i), t^2\$a_1, \dots, t^r\$a_{r-1}, t^{r+1}\$(a_r + a_{r+1})) \right. \\ & \quad + \sum_{i=1}^{a_1} t^{i-1} P_{n-1}^{(r)}(t; 1\$a_0, t\$(i-1), t^2\$(a_1-i), t^3\$a_2, \dots, t^r\$a_{r-1}, t^{r+1}\$(a_r + a_{r+1})) \\ & \quad + \sum_{i=1}^{a_2} t^{a_1+2(i-1)} P_{n-1}^{(r)}(t; 1\$a_0, t\$a_1, t^2\$(i-1), t^3\$(a_2-i), \dots, t^r\$a_{r-1}, t^{r+1}\$(a_r + a_{r+1})) \\ & \quad \quad \quad + \dots \\ & \quad \left. + \sum_{i=1}^{a_{r+1}} t^{a_1+2a_2+\dots+r a_r+(r+1)(i-1)} P_{n-1}^{(r)}(t; 1\$a_0, t\$a_1, \dots, t^r\$a_r, t^{r+1}\$(a_{r+1}-1)) \right]. \end{aligned}$$

Due to the CHOP_r operator, many terms automatically disappear because of the power of t in front. From a computational perspective, this observation eliminates many unnecessary terms and hence circumvents a lot of unnecessary computation. This has been automated in the Maple package **FINCRT** so that a computer can derive a “scheme” for any fixed r (completely on its own) and use it to enumerate $s_n(132, r)$ for as many terms as the user wants.²

For example, the Maple call **F132rN(5, 15)**; for the first 15 terms of $s_n(132, 5)$ produces the sequence:

$$0, 0, 0, 0, 5, 55, 394, 2225, 11539, 57064, 273612, 1283621, 5924924, 27005978, 121861262$$

2.2 Extending to the pattern 1243

First, we outline how to extend the previous approach to the pattern 1243. In addition to the variable t , we now introduce $2n$ catalytic variables x_1, \dots, x_n and y_1, \dots, y_n . The weight of a length n permutation $\pi = \pi_1 \dots \pi_n$ will now be

$$\text{weight}(\pi) := t^{N_{1243}(\pi)} \prod_{i=1}^n x_i^{\#\{(a,b) : \pi_a > \pi_b = i, 1 \leq a < b \leq n\}} \cdot y_i^{\#\{(a,b,c) : \pi_a = i < \pi_c < \pi_b, 1 \leq a < b < c \leq n\}}.$$

For example, $\text{weight}(123456) = 1$ and $\text{weight}(135624) = t^2 x_2^3 x_4^2 y_1^5 y_3^2$.

For each n , we define the polynomial

$$P_n(t; x_1, \dots, x_n, y_1, \dots, y_n) := \sum_{\pi \in S_n} \text{weight}(\pi).$$

Now given a permutation $\pi = \pi_1 \dots \pi_n$, suppose that $\pi_1 = i$. Observe that $N_{1243}(\pi)$ is equal to the number of occurrences of 1243 in $\pi_2 \dots \pi_n$ plus the number of occurrences of 132 in $\pi_2 \dots \pi_n$ where the term corresponding to the “1” is larger than i . Letting $\pi' := \text{red}(\pi_2 \dots \pi_n)$, we observe that

$$\text{weight}(\pi) = x_1 x_2 \dots x_{i-1} \cdot \text{weight}(\pi')|_{x_i \rightarrow y_i x_{i+1}, \dots, x_{n-1} \rightarrow y_i x_n, y_i \rightarrow t y_{i+1}, \dots, y_{n-1} \rightarrow t y_n}. \quad (3)$$

This leads to the functional equation:

$$\begin{aligned} P_n(t; x_1, \dots, x_n, y_1, \dots, y_n) = \\ \sum_{i=1}^n x_1 x_2 \dots x_{i-1} \cdot P_{n-1}(t; x_1, \dots, x_{i-1}, y_i x_{i+1}, \dots, y_i x_n, y_1, \dots, y_{i-1}, t y_{i+1}, \dots, t y_n). \end{aligned} \quad (\text{FE1243})$$

Again, our goal is to compute $f_n(t) = P_n(t; 1 [2n \text{ times}])$. We can apply the same computational methods as before. For example, we can apply (FE1243) directly to $P_n(t; 1 [2n \text{ times}])$ (and more generally, to objects of the form $P_n(t; 1 [a_0 \text{ times}], \dots, t^{s_1} [a_{s_1} \text{ times}], 1 [b_0 \text{ times}], \dots, t^{s_2} [b_{s_2} \text{ times}])$) to compute $f_n(t)$. This again gives us an algorithm that is faster than the direct weighted counting of $n!$ permutations but is still exponential-time (and memory).

This algorithm has been implemented in the procedure **F1243full(n, t)** (in Maple package **FINCRT**). For example, the Maple call **F1243full(8, t)**; computes $f_8(t)$ and outputs:

$$\begin{aligned} &t^{36} + t^{31} + 10t^{30} + 3t^{28} + 13t^{27} + 9t^{26} + 8t^{25} + 37t^{24} + 16t^{23} + 16t^{22} + 49t^{21} + 60t^{20} \\ &+ 41t^{19} + 130t^{18} + 81t^{17} + 157t^{16} + 266t^{15} + 184t^{14} + 233t^{13} + 542t^{12} + 356t^{11} + 771t^{10} \\ &+ 877t^9 + 975t^8 + 972t^7 + 2180t^6 + 1710t^5 + 2658t^4 + 3119t^3 + 4600t^2 + 4478t + 15767 \end{aligned}$$

²The “scheme” mentioned here is a liberal application of the word and differs from *enumeration schemes*.

Additionally, both the obvious analog of Lemma 1 as well as the computational reduction using the CHOP_r operator still apply in this setting. This has also been automated in the Maple package **FINCRT**.

For example, the Maple call `F1243rN(1,15)`; for the first 15 terms of $s_n(1243, 1)$ produces the sequence:

$$0, 0, 0, 1, 11, 88, 638, 4478, 31199, 218033, 1535207, 10910759, 78310579, 567588264, 4152765025$$

and the Maple call `F1243rN(2,15)`; for the first 15 terms of $s_n(1243, 2)$ produces the sequence:

$$0, 0, 0, 0, 4, 56, 543, 4600, 36691, 284370, 2174352, 16533360, 125572259, 955035260, 7283925999$$

2.3 Extending to longer patterns

The approach for the patterns 132 and 1243 can be extended analogously to longer patterns of the form $12\dots(k-2)(k)(k-1)$. For example, if the pattern $\tau = 12354$, we consider the variable t and $3n$ catalytic variables: x_1, \dots, x_n and y_1, \dots, y_n and z_1, \dots, z_n . The weight of a length n permutation $\pi = \pi_1 \dots \pi_n$ will now be

$$\text{weight}(\pi) = t^{N_{12354}(\pi)} \prod_{i=1}^n x_i^{\#\{(a,b) : \pi_a > \pi_b = i\}} \cdot y_i^{\#\{(a,b,c) : \pi_a = i < \pi_c < \pi_b\}} \cdot z_i^{\#\{(a,b,c,d) : \pi_a = i < \pi_b < \pi_d < \pi_c\}}$$

where it is always assumed that $a < b < c < d$.

An analogous functional equation is derived for the corresponding polynomial

$$P_n(t; x_1, \dots, x_n, y_1, \dots, y_n, z_1, \dots, z_n) := \sum_{\pi \in S_n} \text{weight}(\pi)$$

and all the analogous computational methods work in this setting as well. The 12354 case has also been automated in the Maple package **FINCRT**.

For example, the Maple call `F12354rN(0,14)`; for the first 14 terms of $s_n(1243, 0)$ produces the sequence:

$$1, 2, 6, 24, 119, 694, 4582, 33324, 261808, 2190688, 19318688, 178108704, 1705985883, 16891621166$$

and the Maple call `F12354rN(1,15)`; for the first 15 terms of $s_n(1243, 1)$ produces the sequence:

$$0, 0, 0, 0, 1, 19, 246, 2767, 29384, 305646, 3170684, 33104118, 349462727, 3738073247, 40549242195$$

3 Counting occurrences of the pattern $23\dots k1$

In this section, we extend the previous techniques to handle patterns of the form $23\dots k1$. Although $s_n(231, r) = s_n(132, r)$ for every r and n (by reversal), we will develop an approach for handling 231 directly³ and then show how this can be extended to longer patterns of the form $23\dots k1$. This new approach for handling 231 will also be necessary in the next section for considering multiple patterns simultaneously.

³As opposed to computing the equivalent pattern 132.

3.1 Permutations containing 231

In this section, we will assume that our (fixed) pattern $\tau = 231$. We define the analogous polynomial

$$f_n(t) := \sum_{\pi \in \mathcal{S}_n} t^{N_{231}(\pi)}.$$

Recall that the coefficient of t^r in $f_n(t)$ will be exactly $s_n(231, r)$.

In addition to the variable t , we introduce $n(n+1)/2$ catalytic variables $x_{i,j}$ with $1 \leq j \leq i \leq n$ and define the weight of a permutation $\pi = \pi_1 \dots \pi_n$ to be

$$\text{weight}(\pi) := t^{N_{231}(\pi)} \prod_{1 \leq j \leq i \leq n} x_{i,j}^{\#\{(a,b) : \pi_a > \pi_b, \pi_a = i, \pi_b < j, 1 \leq a < b \leq n\}}$$

For example, $\text{weight}(12345) = 1$ and $\text{weight}(24153) = t^2 x_{2,2} x_{4,2} x_{4,3} x_{4,4}^2 x_{5,4} x_{5,5}$.

We will again define an analogous multi-variate polynomial P_n on all the previously defined variables. However, for notational convenience, the $x_{i,j}$ variables will be written as a matrix of variables:

$$X_n := \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \ddots & & \vdots \\ \vdots & x_{i,i} & \vdots \\ x_{n,1} & \cdots & x_{n,n} \end{bmatrix} \quad (4)$$

where we will disregard the entries above the diagonal (i.e., the $x_{i,j}$ entries where $j > i$).

For each n , we now define the polynomial

$$P_n(t; X_n) := \sum_{\pi \in \mathcal{S}_n} \text{weight}(\pi).$$

Recall that $P_n(t; \mathbf{1}) = f_n(t)$, where $\mathbf{1}$ is the matrix of all 1's.

Next, we derive an equation (analogous to Eq. 2 and 3) for this weight function. Given a permutation $\pi = \pi_1 \dots \pi_n$, suppose that $\pi_1 = i$. Observe that $N_{231}(\pi)$ is equal to the number of occurrences of 231 in $\pi_2 \dots \pi_n$ plus the number of occurrences of 21 in $\pi_2 \dots \pi_n$, where the term corresponding to the “2” is greater than i and the term corresponding to the “1” is less than i . We can also make the following two observations. First, in $\text{weight}(\pi)$, the exponents of $x_{k,i}$ and $x_{k,i+1}$ are equal for each k (since $\pi_1 = i$). Second, the number of 231 patterns that include the first term $\pi_1 = i$ is the exponent of $x_{i+1,i}$ plus the exponent of $x_{i+2,i}$ plus ... plus the exponent $x_{n,i}$. Now, let π' be $\text{red}(\pi_2 \dots \pi_n)$, and define the collection of substitutions A to be

$$A := \begin{cases} x_{b,c} \rightarrow x_{b+1,c} & b \geq i, c < i \\ x_{b,c} \rightarrow x_{b+1,c+1} & b \geq i, c > i \\ x_{b,c} \rightarrow t x_{b+1,c} \cdot x_{b+1,c+1} & b \geq i, c = i \end{cases} \quad (5)$$

and observe that

$$\text{weight}(\pi) = x_{i,1}^0 x_{i,2}^1 \dots x_{i,i}^{i-1} \cdot \text{weight}(\pi')|_A. \quad (6)$$

Now, define the operator R_1 on a square matrix X_n and $i < n$ to be:

$$R_1(X_n, i) := \begin{bmatrix} x_{1,1} & \cdots & x_{1,i-1} & tx_{1,i}x_{1,i+1} & x_{1,i+2} & \cdots & x_{1,n} \\ \vdots & \ddots & & \vdots & & & \vdots \\ x_{i-1,1} & & x_{i-1,i-1} & & \cdots & & x_{i-1,n} \\ x_{i+1,1} & \cdots & x_{i+1,i-1} & tx_{i+1,i}x_{i+1,i+1} & x_{i+1,i+2} & \cdots & x_{i+1,n} \\ \vdots & & \vdots & \vdots & \ddots & & \vdots \\ \vdots & & \vdots & \vdots & \ddots & & \vdots \\ x_{n,1} & \cdots & x_{n,i-1} & tx_{n,i}x_{n,i+1} & x_{n,i+2} & \cdots & x_{n,n} \end{bmatrix}. \quad (7)$$

In essence, the R_1 operator deletes the i -th row, merges the i -th and $(i+1)$ -th columns via term-by-term multiplication, and multiplies this new column by a factor of t . If $i = n$, then $R_1(X_n, i)$ is defined to be the $(n-1) \times (n-1)$ matrix obtained by deleting the n -th row and n -th column from X_n .

Observe now that the weight equation (Eq. 6) leads to the functional equation:

$$P_n(t; X_n) = \sum_{i=1}^n x_{i,1}^0 x_{i,2}^1 \dots x_{i,i}^{i-1} \cdot P_{n-1}(t; R_1(X_n, i)). \quad (\text{FE231})$$

Note that while all entries in the matrix are changed for consistency, we will continue to disregard the entries above the diagonal.

Again, our goal is to compute $P_n(t; \mathbf{1})$, and the analogous computational techniques from previous sections will also apply in this setting. For example, we can apply (FE231) directly to $P_n(t; \mathbf{1})$ as opposed to computing $P_n(t; X_n)$ symbolically and substituting $x_{i,j} = 1$ at the end. The following result, which is obvious from the definition of the operator R_1 , provides a substantial simplification:

Lemma 2. *Let A be a square matrix where every row is identical (i.e., the i -th row and the j -th row are equal for every i, j). Then, $R_1(A, i)$ will also be a square matrix with identical rows.*

By Lemma 2, repeated applications of R_1 to the all ones matrix $\mathbf{1}$ will still result in a matrix with identical rows. Therefore, it is sufficient to keep track of only one row as opposed to the entire matrix. Also observe that repeated applications of R_1 to the matrix $\mathbf{1}$ will always result in a matrix whose entries are powers of t . Let $Q_n(t; c_1, \dots, c_n)$ denote the polynomial $P_n(t; C)$, where C is the $n \times n$ matrix where every row is $[c_1, \dots, c_n]$ and every c_i is a power of t . This leads to a functional equation analogous to (FE231):

$$Q_n(t; c_1, \dots, c_n) = \sum_{i=1}^n c_1^0 c_2^1 \dots c_i^{i-1} \cdot Q_{n-1}(t; c_1, \dots, c_{i-1}, tc_i c_{i+1}, c_{i+2}, \dots, c_n). \quad (\text{FE231c})$$

Note that $Q_n(t; 1 [n \text{ times}])$ is exactly our desired polynomial $P_n(t; \mathbf{1}) = f_n(t)$. However, this interpretation only forces us to deal with n catalytic variables (the c_i 's) as opposed to $n(n+1)/2$ catalytic variables (the $x_{i,j}$'s). Just as in prior sections, we can repeatedly apply our functional equation (FE231c) to compute $Q_n(t; 1 [n \text{ times}])$.

When the sequence $s_n(231, r)$ is desired for a fixed r , the obvious analog of Lemma 1 and the computational reduction using the CHOP_r operator can again be used. This has been implemented in the Maple package F231.⁴

⁴Although all output would be equivalent to the 132 case, the approach here will be necessary when considering multiple patterns.

3.2 Extending to the pattern 2341

In this section, we outline how to extend the approach for 231 to an analogous (but more complicated) approach for 2341. In addition to the variable t , we now introduce $n(n+1)/2$ catalytic variables $x_{i,j}$ with $1 \leq j \leq i \leq n$ and $n(n+1)/2$ more catalytic variables $y_{i,j}$ with $1 \leq j \leq i \leq n$ (a total of $n(n+1)$ catalytic variables). Define the weight of a permutation $\pi = \pi_1 \dots \pi_n$ to be

$$t^{N_{2341}(\pi)} \prod_{1 \leq j \leq i \leq n} x_{i,j}^{\#\{(a,b) : \pi_a > \pi_b, \pi_a = i, \pi_b < j, 1 \leq a < b \leq n\}} \cdot y_{i,j}^{\#\{(a,b,c) : \pi_c < \pi_a < \pi_b, \pi_a = i, \pi_c < j, 1 \leq a < b < c \leq n\}}$$

For example, $\text{weight}(24351) = t^2 x_{2,2} x_{3,2} x_{3,3} x_{4,2} x_{4,3} x_{4,4}^2 x_{5,2} x_{5,3} x_{5,4} x_{5,5} y_{2,2}^3 y_{3,2} y_{3,3} y_{4,2} y_{4,3} y_{4,4}$.

The $x_{i,j}$ variables and the $y_{i,j}$ variables will be written as matrices of variables:

$$X_n := \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \ddots & & \\ \vdots & x_{i,i} & \vdots \\ & \ddots & \\ x_{n,1} & \cdots & x_{n,n} \end{bmatrix}, \quad Y_n := \begin{bmatrix} y_{1,1} & \cdots & y_{1,n} \\ \ddots & & \\ \vdots & y_{i,i} & \vdots \\ & \ddots & \\ y_{n,1} & \cdots & y_{n,n} \end{bmatrix} \quad (8)$$

where we will disregard the entries above the diagonal.

For each n , we define the polynomial

$$P_n(t; X_n, Y_n) := \sum_{\pi \in S_n} \text{weight}(\pi)$$

and again $P_n(t; \mathbf{1}, \mathbf{1}) = f_n(t)$ is our desired polynomial.

Next, we derive an equation (analogous to Eq. 2, 3, and 6) for this weight function. Given a permutation $\pi = \pi_1 \dots \pi_n$, suppose that $\pi_1 = i$. Observe that $N_{2341}(\pi)$ is equal to the number of occurrences of 2341 in $\pi_2 \dots \pi_n$ plus the number of occurrences of 231 in $\pi_2 \dots \pi_n$, where the term corresponding to the “2” is greater than i and the term corresponding to the “1” is less than i .

We can also make the following observations. First, in $\text{weight}(\pi)$, the exponents of $x_{k,i}$ and $x_{k,i+1}$ are equal and the exponents of $y_{k,i}$ and $y_{k,i+1}$ are equal for each k (since $\pi_1 = i$). Second, the number of 2341 patterns that include the first term $\pi_1 = i$ is the exponent of $y_{i+1,i}$ plus the exponent of $y_{i+2,i}$ plus \dots plus the exponent $y_{n,i}$. Third, the number of 231 patterns that include the first term $\pi_1 = i$ (i.e., the “2” is equal to i) and whose “1” term is less than k is equal to the exponent of $x_{i+1,k}$ plus the exponent of $x_{i+2,k}$ plus \dots plus the exponent of $x_{n,k}$.

Now, let π' be $\text{red}(\pi_2 \dots \pi_n)$, and define the collection of substitutions A' to be

$$A' := \begin{cases} x_{b,c} \rightarrow y_{i,c} \cdot x_{b+1,c} & b \geq i, c < i \\ x_{b,c} \rightarrow x_{b+1,c+1} & b \geq i, c > i \\ x_{b,c} \rightarrow y_{i,i} \cdot x_{b+1,c} \cdot x_{b+1,c+1} & b \geq i, c = i \\ y_{b,c} \rightarrow y_{b+1,c} & b \geq i, c < i \\ y_{b,c} \rightarrow y_{b+1,c+1} & b \geq i, c > i \\ y_{b,c} \rightarrow t y_{b+1,c} \cdot y_{b+1,c+1} & b \geq i, c = i \end{cases} \quad (9)$$

and observe that

$$\text{weight}(\pi) = x_{i,1}^0 x_{i,2}^1 \dots x_{i,i}^{i-1} \cdot \text{weight}(\pi')|_{A'}. \quad (10)$$

In addition to the previous R_1 operator defined in Eq. 7, we define another operator R_2 on two square matrices X_n and Y_n (of equal dimension) and $i < n$ to be:

$$R_2(X_n, Y_n, i) := \begin{bmatrix} x_{1,1} & \cdots & x_{1,i-1} & y_{i,i}x_{1,i}x_{1,i+1} & x_{1,i+2} & \cdots & x_{1,n} \\ \vdots & \ddots & & \vdots & & & \vdots \\ x_{i-1,1} & & x_{i-1,i-1} & & \cdots & & x_{i-1,n} \\ y_{i,1}x_{i+1,1} & \cdots & y_{i,i-1}x_{i+1,i-1} & y_{i,i}x_{i+1,i}x_{i+1,i+1} & x_{i+1,i+2} & \cdots & x_{i+1,n} \\ \vdots & & \vdots & \vdots & \ddots & & \vdots \\ \vdots & & \vdots & \vdots & \ddots & & \vdots \\ y_{i,1}x_{n,1} & \cdots & y_{i,i-1}x_{n,i-1} & y_{i,i}x_{n,i}x_{n,i+1} & x_{n,i+2} & \cdots & x_{n,n} \end{bmatrix}. \quad (11)$$

If $i = n$, then $R_2(X_n, Y_n, i)$ is defined to be the $(n - 1) \times (n - 1)$ matrix obtained by deleting the n -th row and n -th column from X_n .

Observe that this leads to the functional equation:

$$P_n(t; X_n, Y_n) = \sum_{i=1}^n x_{i,1}^0 x_{i,2}^1 \dots x_{i,i}^{i-1} \cdot P_{n-1}(t; R_2(X_n, Y_n, i), R_1(Y_n, i)). \quad (\text{FE2341})$$

As in prior sections, we recursively apply the functional equation directly to $P_n(t; \mathbf{1}, \mathbf{1})$ (and subsequent instances of P_k). Observe that in this scenario, Lemma 2 still applies for the R_1 operator and more specifically the “ Y_n ” matrix in P_n . While the lemma does not apply to the R_2 operator, this still allows us to reduce the number of catalytic variables. Let $Q_n(t; C, d_1, \dots, d_n)$ denote the polynomial $P_n(t; C, D)$ where every entry of the $n \times n$ matrices C and D are powers of t and every row in D is $[d_1, \dots, d_n]$. We derive an analogous functional equation:

$$Q_n(t; C, d_1, \dots, d_n) = \sum_{i=1}^n c_{i,1}^0 c_{i,2}^1 \dots c_{i,i}^{i-1} \cdot Q_{n-1}(t; R_2(C, D, i), d_1, \dots, d_{i-1}, t d_i d_{i+1}, d_{i+2}, \dots, d_n). \quad (\text{FE2341c})$$

Using this recurrence to compute $Q_n(t; \mathbf{1}, \mathbf{1} [n \text{ times}])$ will yield the desired polynomial $f_n(t)$. This approach allows us to deal with $n(n + 1)/2 + n$ catalytic variables (as opposed to $n(n + 1)$ such variables).

Additionally, for a fixed r , the sequence $s_n(2341, r)$ can be computed by applying Lemma 1 and the CHOP_r operator as necessary. This has been implemented in the procedure **F2341rN(r,N)** (in the Maple package **F2341**).

For example, the Maple call **F2341rN(1,15)**; for the first 15 terms of $s_n(2341, 1)$ produces the sequence:

0, 0, 0, 1, 11, 87, 625, 4378, 30671, 216883, 1552588, 11257405, 82635707, 613600423, 4604595573

and the Maple call **F2341rN(2,15)**; for the first 15 terms of $s_n(2341, 2)$ produces the sequence:

0, 0, 0, 0, 5, 68, 626, 5038, 38541, 289785, 2172387, 16339840, 123650958, 942437531, 7236542705

While we do not present the details here, the same methodology can be applied to longer patterns of the form $23\dots k1$. Analogous functional equations can be derived and used for enumeration.

4 Further extensions

4.1 Tracking inversions

One of the most commonly studied permutation statistic is the inversion number. The inversion number of a permutation $\pi = \pi_1 \dots \pi_n$, denoted by $\text{inv}(\pi)$, is the number of pairs (i, j) such that $1 \leq i < j \leq n$ and $\pi_i > \pi_j$. Equivalently, it is the number of occurrences of the pattern 21 in π . For a (fixed) pattern τ , define the polynomial

$$g_n(t, q) := \sum_{\pi \in S_n} q^{\text{inv}(\pi)} t^{N_\tau(\pi)}. \quad (12)$$

Observe that $g_n(t, 1)$ is exactly $f_n(t)$ from before.

Given a permutation $\pi = \pi_1 \dots \pi_n$, suppose that $\pi_1 = i$. Then, $\text{inv}(\pi)$ is equal to the number of inversions in $\pi_2 \dots \pi_n$ plus the number of elements in π_2, \dots, π_n that are less than i . For any previously defined functional equation, it is enough to insert a q^{i-1} factor in the summation.

For example, if the fixed pattern is $\tau = 132$, the polynomial P_n can be analogously defined as

$$P_n(t, q; x_1, \dots, x_n) := \sum_{\pi \in S_n} q^{\text{inv}(\pi)} \cdot \text{weight}_{132}(\pi)$$

and the analog to functional equation (FE132) would be

$$P_n(t, q; x_1, \dots, x_n) = \sum_{i=1}^n q^{i-1} x_1 x_2 \dots x_{i-1} \cdot P_{n-1}(t, q; x_1, \dots, x_{i-1}, tx_{i+1}, \dots, tx_n).$$

Similarly, the analogous functional equation to (FE231c) would be

$$Q_n(t, q; c_1, \dots, c_n) = \sum_{i=1}^n q^{i-1} c_1^0 c_2^1 \dots c_i^{i-1} \cdot Q_{n-1}(t, q; c_1, \dots, c_{i-1}, tc_i c_{i+1}, c_{i+2}, \dots, c_n).$$

From here, all the previous computational techniques for quick enumeration still apply.

This has been implemented in the procedures `qF123r(n,r,t,q)` and `qF1234r(n,r,t,q)` (in Maple package FINCR), `qF132r(n,r,t,q)` and `qF1243r(n,r,t,q)` (in Maple package FINCRT), `qF231r(n,r,t,q)` (in Maple package F231), and `qF2341r(n,r,t,q)` (in Maple package F2341).

4.2 Counting multiple patterns in permutations

In the preceding sections, various functional equations were derived by considering the first term of a typical permutation⁵ and deriving a recurrence. Given any collection of patterns where such recurrences can be derived, we can also consider those patterns simultaneously.

⁵As opposed to looking at the last term (or the smallest/largest term).

As an example, consider the patterns $\sigma = 123$ and $\tau = 132$. The case of only the pattern 123 was done in [13]. In this setting, the weight of a permutation $\pi = \pi_1 \dots \pi_n$ is defined by

$$\text{weight}_{123}(\pi) := t^{N_{123}(\pi)} \prod_{i=1}^n x_i^{\#\{(a,b) : \pi_a = i < \pi_b, 1 \leq a < b \leq n\}}$$

and the corresponding polynomial is

$$P_n(t; x_1, \dots, x_n) := \sum_{\pi \in S_n} \text{weight}_{123}(\pi).$$

The corresponding functional equation (referred to as the *Noonan-Zeilberger Functional Equation*) is

$$P_n(t; x_1, \dots, x_n) = \sum_{i=1}^n x_i^{n-i} P_{n-1}(t; x_1, \dots, x_{i-1}, tx_{i+1}, \dots, tx_n). \quad (\text{NZFE})$$

This can be merged with the analogous quantities for 132 as follows. Let s and t be the variables corresponding to 123 and 132, respectively. Let x_1, \dots, x_n and y_1, \dots, y_n be two sets of catalytic variables, and define the weight of a permutation $\pi = \pi_1 \dots \pi_n$ by

$$\text{weight}(\pi) := s^{N_{123}(\pi)} t^{N_{132}(\pi)} \prod_{i=1}^n x_i^{\#\{(a,b) : \pi_a = i < \pi_b, 1 \leq a < b \leq n\}} y_i^{\#\{(a,b) : \pi_a > \pi_b = i, 1 \leq a < b \leq n\}}.$$

For each n , we define the polynomial

$$P_n(s, t; x_1, \dots, x_n, y_1, \dots, y_n) := \sum_{\pi \in S_n} \text{weight}(\pi)$$

and can similarly derive the functional equation

$$\begin{aligned} P_n(s, t; x_1, \dots, x_n, y_1, \dots, y_n) = \\ \sum_{i=1}^n x_i^{n-i} y_1 y_2 \dots y_{i-1} P_{n-1}(s, t; x_1, \dots, x_{i-1}, sx_{i+1}, \dots, sx_n, y_1, \dots, y_{i-1}, ty_{i+1}, \dots, ty_n). \end{aligned}$$

The same computational techniques from the prior sections apply here as well.

This has been implemented in the Maple package **F123n132**. For example, the Maple call **F123r132sN(2,2,15)**; gives the first 15 terms of the sequence enumerating permutations with 2 occurrences of 123 and 2 occurrences of 132:

$$0, 0, 0, 1, 6, 26, 94, 306, 934, 2732, 7752, 21488, 58432, 156288, 411904$$

and the Maple call **F123r132sN(4,2,15)**; gives the first 15 terms of the sequence enumerating permutations with 4 occurrences of 123 and 2 occurrences of 132:

$$0, 0, 0, 0, 1, 5, 23, 106, 450, 1740, 6214, 20831, 66427, 203550, 603920$$

Other pairs (or larger sets) of patterns follow similarly, and the analogous 1234 and 1243 case has been implemented in the Maple package **F1234n1243**.

Finally, it is possible to consider all length 3 patterns simultaneously. Only the patterns 123, 132, and 231 were done directly, but analogous functional equations can be derived for 321, 312,

and 213. These six functional equations can be combined to count occurrences of all the length 3 patterns. This has been implemented in the Maple package `FS3`. For example, the Maple call `FS3full(7, [t[1], t[2], t[3], t[4], t[5], t[6]])`; would produce the polynomial

$$\sum_{\pi \in S_7} t_1^{N_{123}(\pi)} t_2^{N_{132}(\pi)} t_3^{N_{213}(\pi)} t_4^{N_{231}(\pi)} t_5^{N_{312}(\pi)} t_6^{N_{321}(\pi)}$$

in its computed and expanded form.⁶

5 Conclusion

In this work, we extended and generalized the techniques of [13] to the pattern families $12\dots(k-2)(k-1)$ and $23\dots k1$. In addition, we showed how this approach could be further extended to handle inversions and more generally, multiple patterns simultaneously. It would be interesting to see what additional patterns this approach can be applied to. Also, while the main results of this paper are enumeration algorithms, they are based off of rigorously derived functional equations. It would be interesting to find out if any additional information can be extracted from these functional equations.

Finally, the techniques of this paper allow us to compute many sequences that lead to new conjectures. Let $c_{r,s}(n)$ be the number of length n permutations with exactly r occurrences of 1234 and exactly s occurrences of 1243. We will denote this as $c(n)$ when r and s is clear from context. When $r = s = 0$, it is known that $c(n)$ is exactly the Schröder numbers. For fixed $r, s \leq 1$, we are almost certain that $c(n)$ is P-recursive⁷ (the algorithm can compute enough terms to guess a recurrence). On the other hand, for the single pattern case, it is not clear if $s_n(1234, 1)$ is P-recursive [13].

Based off of empirical evidence, we also believe the following to be true:

Conjecture 1. *For each fixed $r \geq 0$ and $s \geq 0$, the sequence enumerating length n permutations with exactly r occurrences of 123 and s occurrences of 132 is given by $p(n)2^n$, where $p(n)$ is some polynomial of degree $r + s$.*

There are a number of results considering this type of problem (beginning with [17, 18]), but most such results limit themselves to $s = 0, 1$. If this general form were shown to hold for arbitrary r and s , the `F123n132` package could quickly compute enough terms to find explicit formulas and generating functions.

References

- [1] Andrew Baxter. Refining enumeration schemes to count according to the inversion number. *Pure Math. Appl. (PU.M.A.)*, 21(2):137–160, 2010.
- [2] Andrew Baxter and Lara Pudwell. Enumeration schemes for vincular patterns. *Discrete Math.*, 312(10):1699–1712, 2012.
- [3] Miklós Bóna. A new upper bound for 1324-avoiding permutations. arxiv:1207.2379 [math.co], 2012.

⁶The actual output from Maple is too large to include here. We were able to compute up to `FS3full(11, [t[1], t[2], t[3], t[4], t[5], t[6]])`; which is a 450 megabyte text file.

⁷This is a special case of the Noonan-Zeilberger Conjecture from [15].

- [4] Miklós Bóna. The number of permutations with exactly r 132-subsequences is P -recursive in the size! *Adv. in Appl. Math.*, 18(4):510–522, 1997.
- [5] Miklós Bóna. Permutations with one or two 132 subsequences. *Discrete Math.*, 181(1-3):267–274, 1998.
- [6] Alexander Burstein. A short proof for the number of permutations containing pattern 321 exactly once. *Electron. J. Combin.*, 18(2):Paper 21, 3, 2011.
- [7] David Callan. A recursive bijective approach to counting permutations containing 3-letter patterns. arxiv:math/0211380 [math.co], 2002.
- [8] Anders Claesson, Vít Jelínek, and Einar Steingrímsson. Upper bounds for the Stanley-Wilf limit of 1324 and other layered patterns. *J. Combin. Theory Ser. A*, 119(8):1680–1691, 2012.
- [9] Markus Fulmek. Enumeration of permutations containing a prescribed number of occurrences of a pattern of length three. *Adv. in Appl. Math.*, 30(4):607–632, 2003.
- [10] Sergey Kitaev and Toufik Mansour. A survey on certain pattern problems. *University of Kentucky research report 2003-09*, 2003.
- [11] Donald E. Knuth. *The art of computer programming*. Vol. 1: Fundamental algorithms. Addison Wesley, Reading, Massachusetts, 1973.
- [12] Toufik Mansour and Alek Vainshtein. Counting occurrences of 132 in a permutation. *Adv. in Appl. Math.*, 28(2):185–195, 2002.
- [13] Brian Nakamura and Doron Zeilberger. Using Noonan-Zeilberger functional equations to enumerate (in polynomial time!) generalized Wilf classes. *Adv. in Appl. Math.*, to appear.
- [14] John Noonan. The number of permutations containing exactly one increasing subsequence of length three. *Discrete Math.*, 152(1-3):307–313, 1996.
- [15] John Noonan and Doron Zeilberger. The enumeration of permutations with a prescribed number of “forbidden” patterns. *Adv. in Appl. Math.*, 17(4):381–407, 1996.
- [16] Lara Pudwell. Enumeration schemes for permutations avoiding barred patterns. *Electron. J. Combin.*, 17(1):Research Paper 29, 27, 2010.
- [17] Aaron Robertson. Permutations containing and avoiding 123 and 132 patterns. *Discrete Math. Theor. Comput. Sci.*, 3(4):151–154 (electronic), 1999.
- [18] Aaron Robertson, Herbert S. Wilf, and Doron Zeilberger. Permutation patterns and continued fractions. *Electron. J. Combin.*, 6:Research Paper 38, 6 pp. (electronic), 1999.
- [19] Rodica Simion and Frank W. Schmidt. Restricted permutations. *European J. Combin.*, 6(4):383–406, 1985.
- [20] Vince Vatter. Enumeration schemes for restricted permutations. *Combin. Probab. Comput.*, 17(1):137–159, 2008.
- [21] Doron Zeilberger. Enumeration schemes and, more importantly, their automatic generation. *Ann. Comb.*, 2(2):185–195, 1998.